Amendments to the Specification:

Please replace page 13, line 27 to page 14 line 15 with the following amended lines:

The current embodiment of the invention uses a four line communication interface and method of communicating between the FPGA within base station 218 (acting as a "virtual microcontroller" 220 or ICE) and the real microcontroller device under test (microcontroller 232). The four line communication interface is time-dependent so that different information can be transferred at different times over a small number of communication lines. Moreover, since the two processors operate in lockstep, there is no need to provide bus arbitration, framing, or other protocol overhead to effect the communication between the microcontroller 232 and the virtual microcontroller 220. This interface is used for, among other things, transferring of I/O data from the microcontroller 232 to the FPGA 220 (since the FPGA emulates only the core processor functions of the microcontroller in the current embodiment). A first interface line [[(Data1)]] (Data0) is a data line used by the microcontroller 232 to send I/O data to the FPGA based virtual microcontroller 220. This line is also used to notify the FPGA 220 of pending interrupts. This [[Data1]] Data0 line is only driven by the real microcontroller 232. A second data line [[(Data2)]] Data1, which is bidirectional, is used by the microcontroller 232 to send I/O data to the FPGA based virtual microcontroller of base station 218. In addition, the FPGA 220 uses the [[Data2]] Data1 line to

convey halt requests (i.e., to implement simple or complex breakpoints) to the microcontroller 232.

Please replace page 16 Table 1 with the newly amended Table 1.

Please replace page 19 line 9-22 with the following amended lines:

At the end of the four clock cycle data transfer phase in the current embodiment, the control phase 318 begins. During this control phase, which in the current embodiment may be as short as two microcontroller clock periods (or as long as about fourteen clock periods, depending upon the number of cycles required to execute an instruction), the microcontroller 232 can send interrupt requests, interrupt data, and watchdog requests. Additionally, the virtual microcontroller 220 can issue halt (break) commands. If a halt command is issued it is read by the microcontroller at the next SOI signal. Once the control phase ends, the data transfer phase repeats. If there is no data to transfer, [[data1]] data0 and [[data2]] data1 remain idle (e.g., at a logic low state). To simply the circuitry, I/O bus data are sent across the interface on every instruction, even if it is not a bus transfer. Since the virtual microcontroller 220 is operating in synchronization with microcontroller 232 and executing the same instructions, the emulation system knows that data transferred during non I/O red transfers can be ignored.

Please replace page 20 line 20 to page 21 line 13 with the following amended lines:

A transfer of I/O data as just described is illustrated with reference to the timing diagram of FIGURE 5. After the microcontroller 232 completes an I/O read instruction, it sends the read data back to the base station 218 to the virtual microcontroller, since the virtual microcontroller 220 of the present embodiment implements only the core processor functions (and not the I/O functions). The ICE system can expect the incoming data stream for an I/O read to commence with the first positive edge of U_HCLK (the debug or system clock) when SOI signal for the following instruction is at a predetermined logic level (e.g., a logic high). Thus, at time T1, the SOI signal makes a transition to a logic high and one system clock cycle later at time T2, the data transfer phase 310 begins. This timing allows the ICE system to get the read data to the emulated accumulator of base station 218 before it is needed by the next instruction's execution. Note that the first SOI pulse shown in FIGURE 5 represents the first SOI following the I/O read instruction (but could be any suitable reference time signal). Transfer of the data from the microcontroller 232 is carried out using the two datalines ([[data2]] data1 and [[data1]] data0, as shown as U_D0_BRK and U_D1_IRQ) with each line carrying four bits of an eight bit word. During this data transfer phase 310, an eight bit transfer representing the I/O read data can take place from the

microcontroller 232 to the base station [[210]] 218 in the four clock cycles between T2 and T3. The control phase 318 starts at time T3 and continues until the beginning of the next data transfer phase 310. The SOI signal at T4 indicates that the next data transfer phase is about to start and serves as a reference time to read the [[data2]] data1 line to detect the presence of any halt signal from the virtual microcontroller 220. The control phase 318 ends at T5 and the next data transfer phase 310 begins.

Signal Name	Signal Direction with Respect to Base Station 218	Description
U_HCLK (Data Clock or HCLOCK)	In	24/48MHz data clock driven by microcontroller 232. This clock is used to drive the ICE virtual microcontroller communication state machines. This clock always runs at 24MHz, unless the U_CCLK clock is running at 24MHz — then it switches to 48MHz.
U_CCLK (microcontroller Clock or CCLOCK)	In	The internal microcontroller 232 CPU clock.
U_D1_IRQ (Data0)	In	One of two data lines used by the microcontroller 232 to send I/O data to the ICE. This line is also used to notify the ICE of pending interrupts. This line is only driven by the microcontroller 232 (i.e., unidirectional).
U_D0_BRQ (Data1)	In/Out	One of two data lines used by the microcontroller 232 to send I/O data to the ICE. The ICE uses this line to convey halt requests and other information to the microcontroller 232. This line is used for bi-directional communication.
ICE_POD_RST (RESET)	Out	Optional active high reset signal to microcontroller 232.
ICE_POD_PW_R (POWER)	Out	Optional power supply to microcontroller 232.
ICE_POD_GND (GROUND)	Out	Optional ground wire to microcontroller 232.

TABLE 1